Physical Oceanography Distributed Active Archive Center

New examples of encoding satellite geolocation information

Edward Armstrong[1], Sasha Ignatov[2], Matt Pennybaker[2], and Wen-Hao Li[1]

1. Jet Propulsion Laboratory, California Institute of Technology

2. NOAA STAR.

Implementations of novel geolocation encoding for Level 1and 2 satellite data

- Implementation #1
  - Encoding geolocation for the geostationary GOES-16 (aka “GOES R”) platform
    - GOES ABI (sensor) for atmospheric and oceanographic observation
    - New technique leveraging CF projection information for the NOAA GHRSST L2P SST dataset
      - https://podaac.jpl.nasa.gov/dataset/ABI_G16-STAR-L2P-v2.70
- Implementation #2
  - Encoding geolocation for polar orbiting NPP VIIRS sensor
    - Multispectral VIIRS used for all manner of earth atmosphere, land and ocean observation. MODIS follow-on.
    - Eumetsat has developed pixel level interpolation technique to retrieve subsampled L1 geolocation points

Leveraging CF for geostationary SST datasets

- The Group for High Resolution SST (GHRSST) has a specific data model for encoding Level 2 geolocation latitude and longitude
  - Every pixel is assigned its unique lat/lon
  - lat/lon information stored as floats; “heaviest” variables in the granule suite
  - See the GDS reference documentation
    - https://podaac-tools.jpl.nasa.gov/drive/files/allData/ghrsst/docs/GDS20r5.pdf
- The geostationary view of the observation “disk”, essentially a grid, lends itself to encoding using a map projection which is possible using the “grid_mapping” CF metadata attribute
- How does this look like and perform? …….

Existing L2P GHRSST encoding

- Example NetCDF CDL from previous version of the GOES-16 dataset

```
dimensions:
        ni = 5424 ;
        nj = 5424 ;
        time = 1 ;

float lat(nj, ni) ;
                lat:standard_name = "latitude" ;
                lat:units = "degrees_north" ;
                lat:valid_max = 90.f ;
                lat:valid_min = -90.f ;
float lon(nj, ni) ;
                lon:standard_name = "longitude" ;
                lon:units = "degrees_east" ;
                lon:valid_max = 180.f ;
                lon:valid_min = -180.f ;
short sea_surface_temperature(time, nj, ni) ;
                sea_surface_temperature:add_offset = 273.15f ;
                sea_surface_temperature:coordinates = "lon lat" ;
                sea_surface_temperature:scale_factor = 0.01f ;
                sea_surface_temperature:standard_name = "sea surface sub-skin temperature" ;
                sea_surface_temperature:units = "kelvin" ;
```

New L2P GHRSST encoding

- Example NetCDF CDL from new version of the GOES-16 dataset

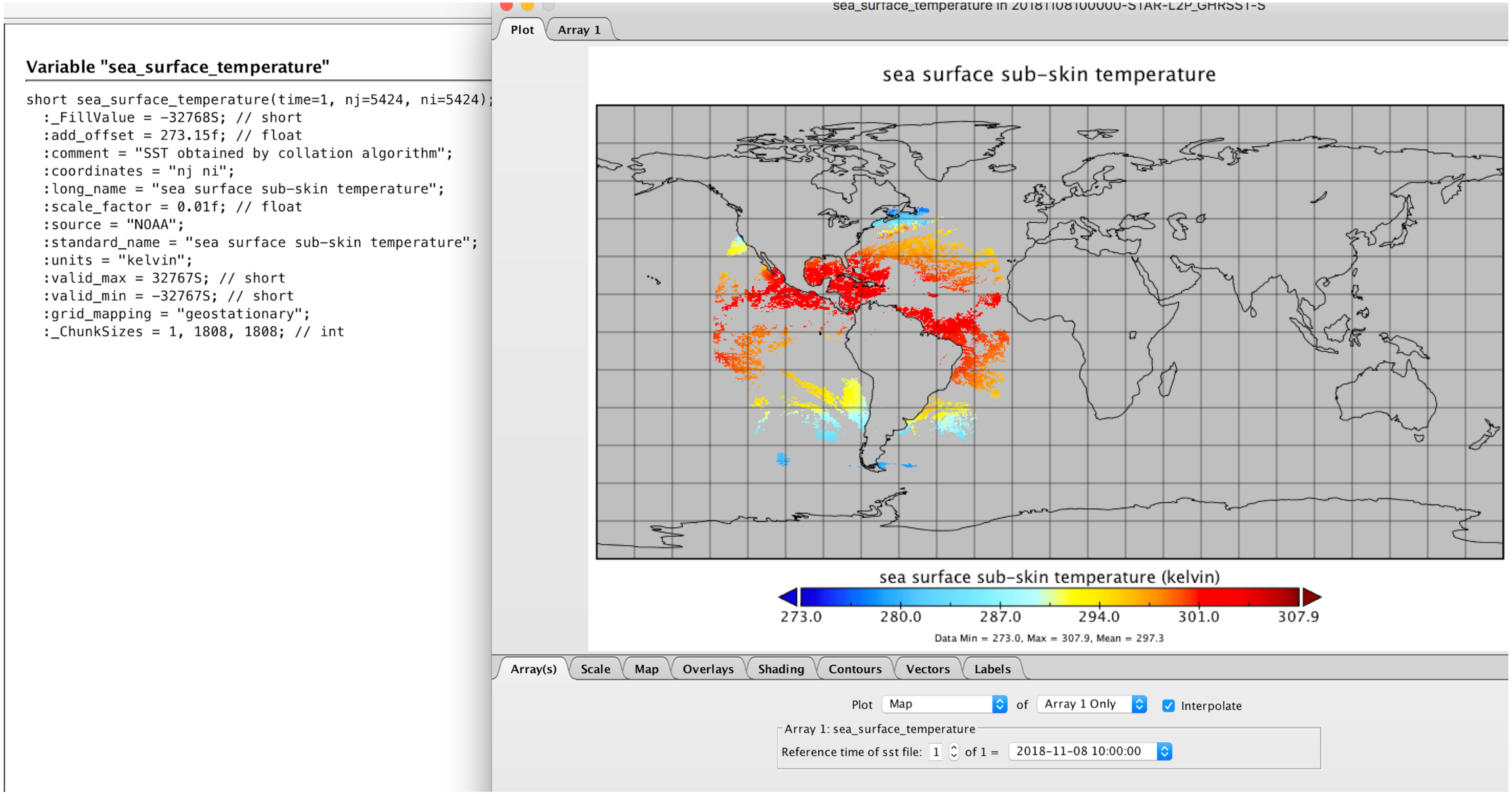
```
dimensions:
        time = 1 ;
        nj = 5424 ;
        ni = 5424 ;

float nj(nj) ;
                nj:axis = "Y" ;
                nj:units = "radians" ;
                nj:standard_name = "projection_y_coordinate" ;
float ni(ni) ;
                ni:axis = "X" ;
                ni:units = "radians" ;
                ni:standard_name = "projection_x_coordinate" ;
short sea_surface_temperature(time, nj, ni) ;
                sea_surface_temperature:add_offset = 273.15f ;
                sea_surface_temperature:coordinates = "nj ni" ;
                sea_surface_temperature:scale_factor = 0.01f ;
                sea_surface_temperature:standard_name = "sea surface sub-skin temperature" ;
                sea_surface_temperature:units = "kelvin" ;
                sea_surface_temperature:grid_mapping = "geostationary" ;
```

CF projection information

- NetCDF CDL from new version of the GOES-16 dataset

```
int geostationary ;
        geostationary:grid_mapping_name = "geostationary" ;
        geostationary:semi_major_axis = 6378137. ;
        geostationary:semi_minor_axis = 6356752.314245 ;
        geostationary:inverse_flattening = 298.257223563 ;
        geostationary:latitude_of_projection_origin = 0. ;
        geostationary:longitude_of_projection_origin = -75. ;
        geostationary:false_easting = 0. ;
        geostationary:false_northing = 0. ;
        geostationary:horizontal_datum_name = "WGS_1984" ;
        geostationary:reference_ellipsoid_name = "WGS 84" ;
        geostationary:prime_meridian_name = "Greenwich" ;
        geostationary:geographic_crs_name = "WGS 84" ;
        geostationary:sweep_angle_axis = "x" ;
        geostationary:perspective_point_height =
```

Plotting with Panoply



Granule data reduction → greater than 80% !!

- Average: 272 MB → 44 MB

PO.DAAC Drive

Back

Logged in as edward.m.armstrong Logout ?

Current Location:

files / OceanTemperature / ghrsst / data / GDS2 / L2P / GOES16 / OSPO / v2.5 / 2018 / 123 /

Access PO.DAAC Drive API Credentials

Name	Last Modified	Size
Parent Directory	-	-
20180503000040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc	2018-05-03 01:34:53	272.9 MB
20180503000040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc.md5	2018-05-03 01:34:57	111 Bytes
20180503010040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc	2018-05-03 02:32:01	272.2 MB
20180503010040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc.md5	2018-05-03 02:32:04	111 Bytes
20180503020040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc	2018-05-03 03:14:45	271.8 MB
20180503020040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc.md5	2018-05-03 03:14:48	111 Bytes
20180503030040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc	2018-05-03 03:58:39	271.4 MB
20180503030040-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc.md5	2018-05-03 03:58:42	111 Bytes
20180503040041-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.50-v02.0-fv01.0.nc	2018-05-03 04:53:22	271.5 MB

PO.DAAC Drive

Back

Logged in as edward.m.armstrong Logout ?

Current Location:

files / OceanTemperature / ghrsst / data / GDS2 / L2P / GOES16 / STAR / v2.70 / 2018 / 123 /

Access PO.DAAC Drive API Credentials

Name	Last Modified	Size
Parent Directory	-	-
20180503000000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc	2019-05-17 08:53:24	45.5 MB
20180503000000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc.md5	2019-05-17 08:53:21	112 Bytes
20180503010000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc	2019-05-17 08:55:12	44.8 MB
20180503010000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc.md5	2019-05-17 08:55:09	112 Bytes
20180503020000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc	2019-05-17 08:51:49	44.2 MB
20180503020000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc.md5	2019-05-17 08:51:49	112 Bytes
20180503030000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc	2019-05-17 08:52:26	43.5 MB
20180503030000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc.md5	2019-05-17 08:52:26	112 Bytes
20180503040000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc	2019-05-17 08:52:47	42.9 MB

erature/ghrsst/data/GDS2/L2P/GOES16/STAR/v2.70/2018/123/20180503010000-STAR-L2P_GHRSST-SSTsubskin-ABI_G16-ACSPO_V2.70-v02.0-fv01.0.nc

Eumetsat L2 VIIRS compact data model

EUMETSAT

Compact VIIRS SDR Product Format User Guide

This Document is Public

Doc.No.	:	EUM/TSS/DOC/13/708025
Issue	:	v1I e-signed
Date	:	25 January 2018
WBS/DBS	:	

EUMETSAT
Eumetsat-Allee 1, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555
http://www.eumetsat.int

VIIRS Compact data model

- **1 INTRODUCTION**
  1.1 Purpose
  During the implementation of the EUMETSAT provided VIIRS Regional Service (EARS-VIIRS) a need was identified to develop a Compact VIIRS SDR Product Format (Level 1) to achieve a **cost efficient distribution** of the VIIRS data via EUMETCast, EUMETSAT's satellite based data distribution system.

  This document specifies the Compact VIIRS SDR Product Format and how it relates to the Original VIIRS SDR Product Format developed as part of the Suomi-NPP and JPSS Programmes. It provides guidelines on how to construct the Compact product format from the Original product format and on how to reconstruct the Original product format from the Compact product format.

VIIRS Compact data model

- In the Compact VIIRS SDR format, geolocation data is stored only for the corner points, i.e. the Tie-Points, of each Tie-Point Zone. Interpolation functions are defined for re-constructing the geolocation data for all pixels within the Tie-Point Zone.

Original layout

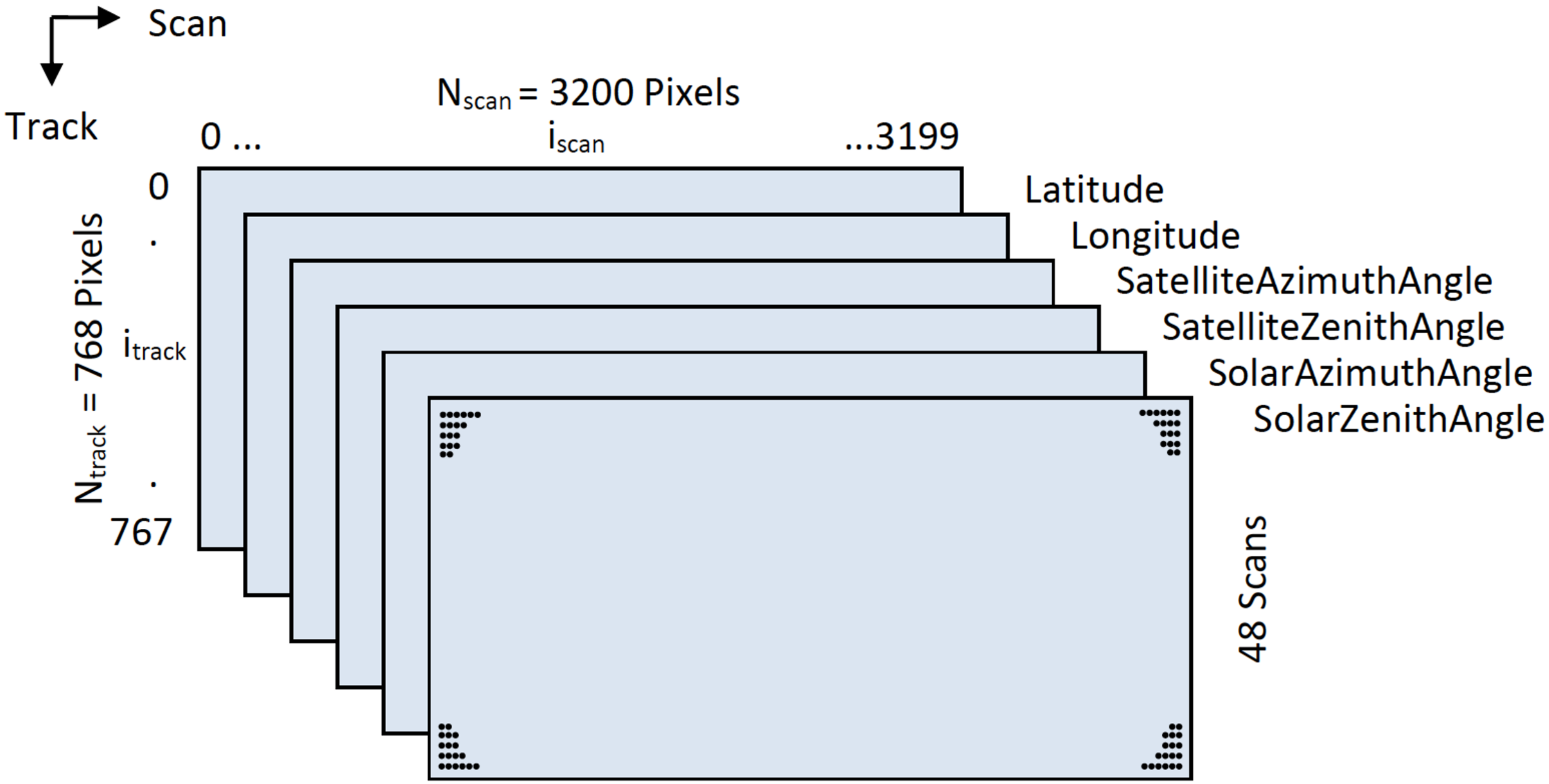


Figure 4 Layout of the Geolocation data in the original VIIRS SDR Product, based on the example of one granule of the VIIRS M-Band

Tie point example

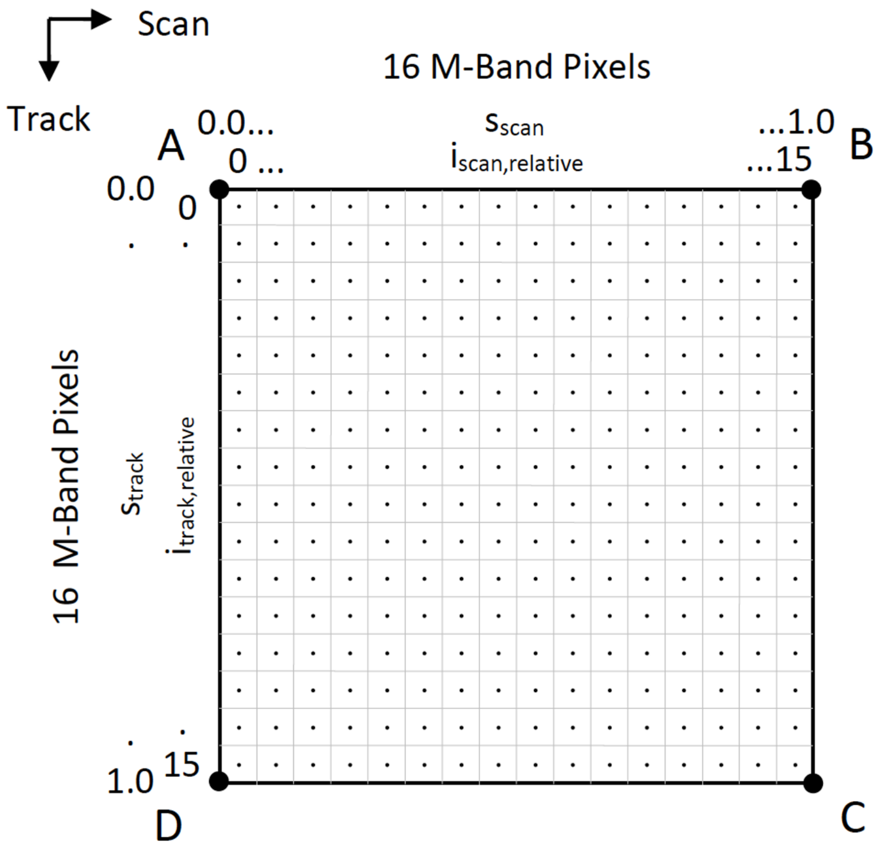


Figure 5 Tie Point Zone Layout. The Compact VIIRS SDR Product stores the six geolocation and angular parameters only in the four corner points A, B, C and D.

The Tie Point grid

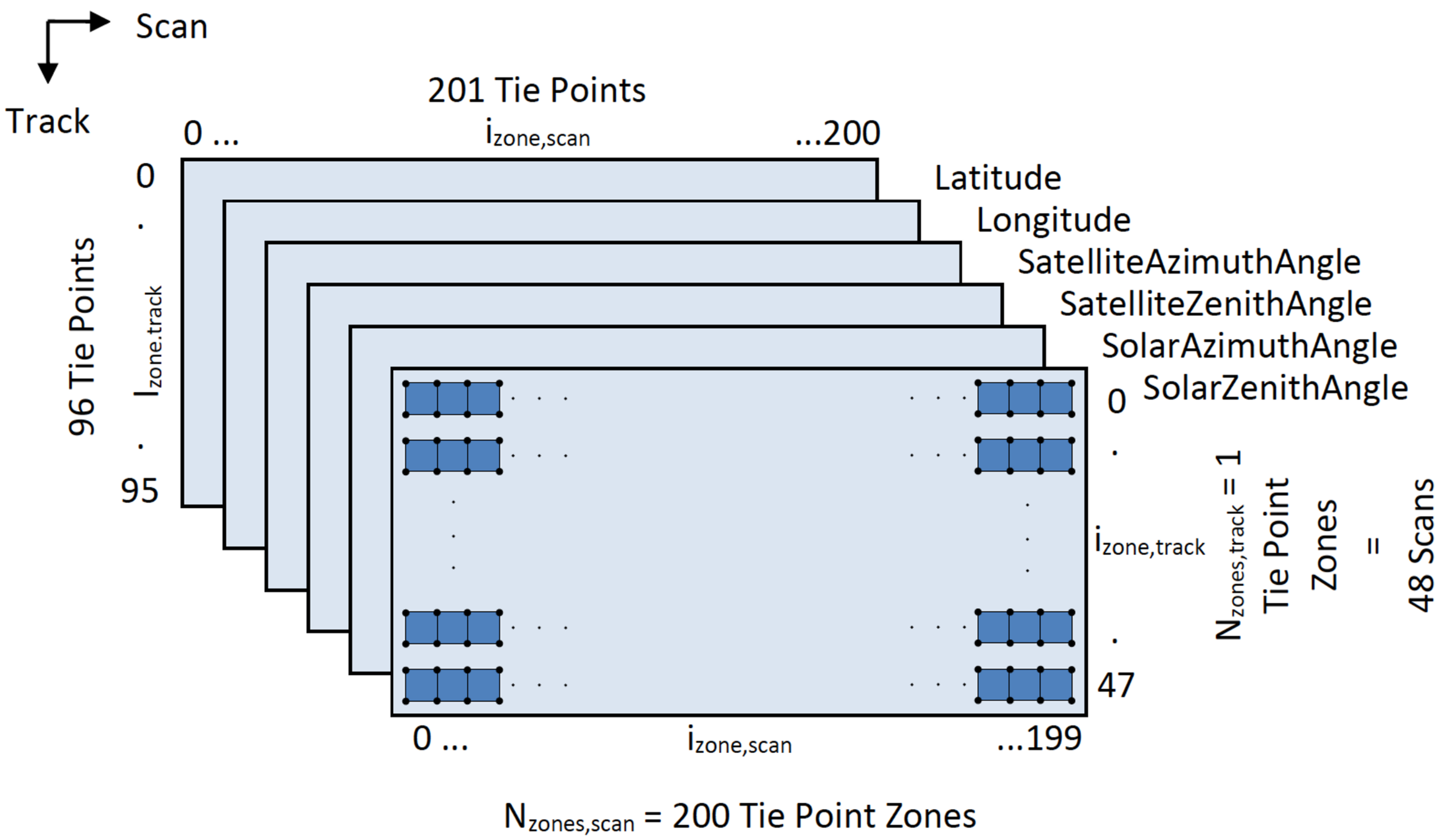


Figure 10 Geolocation and Angular parameter Layout in the Compact VIIRS SDR Product for the M- and I-Band.

Reconstruction of Original Longitude and Latitude

- Each Tie Point set, with lon/lat for corners A,B,C,D and alpha representing satellite zenith angle

10.12.3 Longitude, Latitude Interpolation/Extrapolation

Within a Tie Point Zone, a latitude and longitude can be interpolated based on the Tie Points A, B, C and D as well as the corrected interpolation parameters α_{track} and α_{scan} for the pixel

$$\begin{pmatrix} lat_1 \\ lon_1 \end{pmatrix} = (1 - \alpha_{scan}) \begin{pmatrix} lat_A \\ lon_A \end{pmatrix} + \alpha_{scan} \begin{pmatrix} lat_B \\ lon_B \end{pmatrix}$$

$$\begin{pmatrix} lat_2 \\ lon_2 \end{pmatrix} = (1 - \alpha_{scan}) \begin{pmatrix} lat_D \\ lon_D \end{pmatrix} + \alpha_{scan} \begin{pmatrix} lat_C \\ lon_C \end{pmatrix}$$

$$\begin{pmatrix} lat \\ lon \end{pmatrix} = (1 - \alpha_{track}) \begin{pmatrix} lat_1 \\ lon_1 \end{pmatrix} + \alpha_{track} \begin{pmatrix} lat_2 \\ lon_2 \end{pmatrix}$$

Summary

- CF provides an elegant roadmap for encoded geolocation geostationary L1 and L2 by leveraging projection information similar to an L3 grid
  - 80% data reduction
- Eumetsat has developed a data model to decimate and then interpolate back the geolocation information for L1 polar orbiting satellites
  - Substantial size reduction. Probably an order of magnitude
  - But complicated to retrieve original geolocation values
  - Have any software developers tackled this ?
- What else is on the horizon ?